



Nouveaux algorithmes pour la propagation de contraintes temporelles

Hany Tolba, François Charpillet, Jean-Paul Haton, Gérald Masini

► To cite this version:

Hany Tolba, François Charpillet, Jean-Paul Haton, Gérald Masini. Nouveaux algorithmes pour la propagation de contraintes temporelles. [Rapport de recherche] RR-1458, INRIA. 1991, pp.17. inria-00075103

HAL Id: inria-00075103

<https://inria.hal.science/inria-00075103>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

UNITÉ DE RECHERCHE
INRIA-LORRAINE

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

N° 1458

Programme 3

*Intelligence artificielle, Systèmes cognitifs et
Interaction homme-machine*

NOUVEAUX ALGORITHMES POUR LA PROPAGATION DE CONTRAINTES TEMPORELLES

Hany TOLBA
François CHARPILLET
Jean-Paul HATON
Gérald MASINI

Juin 1991



Nouveaux algorithmes pour la propagation de contraintes temporelles¹

Hany TOLBA, François CHARPILLET,
Jean-Paul HATON, Gérald MASINI.
CRIN/INRIA-Nancy, BP 239, 54506 Vandœuvre, France

résumé :

Le temps joue un rôle notable dans le raisonnement lié à de nombreux domaines d'activité. Nous travaillons d'une part à la représentation du temps et à son utilisation dans le raisonnement. Nous cherchons notamment une représentation permettant de combiner les notions d'intervalles, d'instant et de durée. Dépendant de la complétude des informations temporelles dont on dispose, une représentation numérique ou symbolique du temps est utilisée. L'exploitation d'une telle représentation dans un raisonnement s'appuie sur un ensemble d'outils fondés notamment sur la technique des gestionnaires de graphes temporels (TMM). Dans ce cadre nous avons développé un ensemble d'algorithmes de propagation de contraintes qui fonctionnent soit sur une représentation symbolique (par une extension de GAC4) soit une représentation numérique et discrète (ANGEL) du temps.

abstract :

Time plays an important role especially in reasoning. In this research work we study temporal information representation and how it can be exploited in the reasoning domain. We propose a time representation based on combining the notions of intervals, dates and durations. The manipulation of our representation scheme is based on the notion of time map managers (TMM). We have developped a constraints propagation toolbox which works either on symbolic (by extending GAC4) or a numerical (ANGEL) time representation.

N° de programme INRIA : 3 (Intelligence artificielle, sciences cognitives et interaction
homme-machine)

¹ Cet article a été présenté aux 11èmes Journées Internationales sur les Systèmes Experts d'Avignon en mai 1991 et obtenu le prix INGENIA de la meilleure communication.

1 Cadre De Travail

L'importance du temps et sa représentation ont attiré l'attention des chercheurs en IA ces vingt dernières années. Beaucoup de travaux ont été menés pour aboutir à une représentation générale du temps ainsi qu'à des outils qui permettent l'utilisation de celle-ci. Comme exemples d'outils on peut citer la propagation de contraintes [Mon74], et les systèmes de maintien de vérité [Doy79],[DeK86].

Nous nous intéressons dans ce cadre à la propagation de contraintes, sa problématique dans le domaine du *raisonnement temporel* et nous présenterons les solutions que nous avons développées au CRIN-INRIA Lorraine. Nous présentons aussi le travail en cours, en vue d'une représentation générale du temps.

Dans une première partie nous abordons l'état de l'art dans notre domaine de travail. Nous y présentons les représentations les plus répandues et leurs techniques, en insistant sur la propagation de contraintes et sa problématique.

Dans une deuxième étape nous présentons les solutions que nous proposons. Dans la troisième partie nous détaillons nos algorithmes et nous présentons leurs résultats.

1.1 Etat de l'art

Diverses approches ont été proposées par les chercheurs pour mieux comprendre et représenter le temps. Nous pouvons distinguer deux méthodes qui ont été adoptées : la première est une approche purement numérique et fondée sur la *recherche opérationnelle* (RO); la seconde est une approche fondée sur la logique.

Les chercheurs de la première école proposent une représentation sous forme d'une droite temporelle, sur laquelle ils cherchent à situer numériquement les événements. Ces événements sont représentés sous la forme de fenêtres temporelles, ou de dates au plus tôt et au plus tard. De telles représentations emploient des méthodes de RO pour gérer les relations temporelles: méthode du chemin critique ([Tat77], [Ver83] et [BT84]) ou programmation linéaire [MB83].

La deuxième école repose sur la représentation symbolique et numérique du temps fondée sur la logique. On peut distinguer trois approches relevant

de la logique classique, de la logique modale et de la logique réifiée. C'est à cette dernière que nous nous intéressons dans la suite.

En se fondant sur cette logique, nous pouvons distinguer deux formalismes du temps: le temps subjectif et le temps objectif que nous appellerons également respectivement temps de haut niveau ([Sho87], [KS86]) et temps de bas niveau [Rit88].

Les travaux se partagent entre ces deux formalismes. Le premier recouvre l'aspect causal et irréversible des phénomènes inscrits dans le temps et a été abordé sous un angle formel. Le plus célèbre exemple d'un tel formalisme est le travail de McDermott [McD82] sur l'algèbre des instants. Un fait F est défini, dans cette logique, par l'ensemble des instants t pendant lesquels il est vrai ce qui se note *vrai(fait, t)*. Notons, le statut spécial qui est donné au temps dans cette logique. Dans la logique de McDermott l'ensemble des instants est dense, partiellement ordonné pour la relation de précédence vers le futur et linéaire sur le passé. Les seules relations possibles sur une ramification de la ligne temporelle sont les relations de précédence (\prec , \succ , $=$). Deux instants ordonnés définissent un intervalle appelé chronique lorsqu'il est non borné. Un événement est un ensemble d'intervalles. Les avantages d'une telle représentation reposent sur la capacité à définir les dates et les durées des événements. Par contre, tirer une définition des faits à partir de l'ensemble des instants a des effets graves, notamment l'assimilation entre un fait et l'ensemble des instants pendant lesquels il est vrai est une limitation importante. Il n'est donc pas possible de distinguer dans ce formalisme deux faits ayant la même occurrence. Enfin il n'est possible de considérer qu'une seule ligne temporelle choisie parmi l'ensemble des ramifications vers le futur.

L'approche "temps objectif" est plus descriptive et se traduit le plus souvent par la construction de *gestionnaires de graphes temporels (TMM)*. Le but est de gérer un tel graphe et de le modifier pour aboutir d'une certaine manière à une interprétation correspondant le plus exactement possible au monde réel. Citons par exemple le travail de Dean [DM87] qui s'appuie sur la logique des instants de McDermott pour réaliser un gestionnaire de base de données temporelles. Notons qu'il s'agit d'une des premières approches intégrant un système de maintien de vérité. Citons également les travaux de Allen [All83] qui est sûrement le plus célèbre dans ce domaine. La représentation de Allen est connue sous le nom d'algèbre des intervalles. Dans cette représentation, Allen a défini six relations mutuellement exclusives et leurs inverses qui bénéficient de la même propriété, ainsi que la relation

d'égalité. Celles-ci définissent tous les positionnements possibles entre deux entités.

Allen a défini une règle de transitivité qui permet de déduire la position relative entre deux entités temporelles en connaissant la relation de chacune d'elles avec une troisième. Cette règle est fondée sur une loi de composition. Grâce à cette règle, Allen a implanté un tableau (Tableau de Transitivité) avec lequel il peut déduire la relation entre trois entités temporelles en appliquant la règle de transitivité pour ses treize relations. Cette représentation possède des propriétés très intéressantes, notamment sa capacité à exprimer toutes les relations mutuellement exclusives entre deux entités temporelles. Un autre aspect est qu'il n'y a pas de relation incluse dans une autre. Cependant cette représentation possède des inconvénients. Ainsi il n'est pas possible de définir ni la durée ni la date d'une entité temporelle. Par ailleurs la vérification d'un ensemble de relations entre intervalles est un problème NP-complet [VK86]. Allen a défini un réseau de contraintes (on peut voir le graphe comme un réseau de contraintes) où les nœuds sont les intervalles (événements) et les arcs les relations temporelles (qui sont des relations symboliques). En utilisant la propagation de contraintes, Allen a pu garder la cohérence de son réseau, et propager les effets de toutes les modifications faites localement sur le réseau. Cette propagation utilise le tableau de transitivité pour déduire toutes les relations possibles entre les intervalles et en même temps élimine toutes les relations qui rendraient le réseau inconsistant.

La propagation de contraintes est une technique souvent utilisée avec les différentes représentations du temps de bas niveau. C'est pourquoi nous nous sommes intéressés à analyser la problématique d'une telle technique et à proposer des solutions aux problèmes qui lui sont liées.

Le paragraphe suivant introduit les problèmes dont les représentations de bas niveau souffrent, ainsi que les différentes solutions existantes et leurs limites.

1.2 Propagation de contraintes

Beaucoup de définitions ont été données pour la propagation de contraintes (connue sous le nom CSP). Elles correspondent à des aspects si divers qu'il est difficile d'en donner une définition unique et formelle. Nous présentons une définition que nous pensons correspondre à nos besoins.

Soit N l'ensemble des nœuds n , L l'ensemble des étiquettes l , et $E =$

$\{(i, j) \in N \times N\}$ l'ensemble des arcs entre les nœuds. Généralement sur un graphe, les nœuds sont des objets, les étiquettes représentent le domaine de la valeur d'un objet, et les arêtes définissent les relations entre ces objets. Le problème consiste alors à trouver les étiquettes admissibles parmi les étiquettes possibles pour chaque objet en respectant les relations (exprimant une sorte de compatibilité) entre les objets.

La technique de propagation de contraintes est largement utilisée par les chercheurs en raisonnement temporel. Ici, et comme dans le domaine de la représentation des informations temporelles, la tendance d'utiliser de deux façons différentes la propagation de contraintes: la propagation de contraintes symboliques, et la propagation de contraintes numériques.

1.2.1 La propagation de contraintes symboliques

Ce type de système (représentation et propagation de contraintes) a été introduit par Allen [All83].

Il utilise un algorithme de propagation de contraintes que l'on peut définir comme un cas particulier d'un algorithme de consistance de chemins. Il a défini un graphe dont les nœuds sont les intervalles et les arêtes les relations temporelles. Cet algorithme travaille sur le graphe entier afin d'expliciter toutes les relations entre tous les nœuds. L'algorithme *n'autorise que l'ajout* de relations qui précisent le graphe en éliminant les relations impossibles entre deux intervalles. Pour des questions de complexité, l'algorithme de Allen ne vérifie la consistance qu'entre trois sommets du graphe (cf figure 1). Il n'assure donc pas la consistance globale du graphe. Grâce à cette simplification l'algorithme est de complexité en n^3 où n est le nombre de nœuds du graphe. [VK86].

Van Beek [vB89] propose un algorithme similaire qui permet d'assurer une consistance locale à 4 sommets. La complexité en n^4 est cependant plus élevée.

1.2.2 La propagation de contraintes numériques

Les systèmes numériques manipulent explicitement des nombres et utilisent l'aspect métrique [Rit86][Rit88]. Rit fournit un modèle théorique pour la définition et la propagation des contraintes temporelles numériques. Les objets de base sont des événements qui sont contraints numériquement par

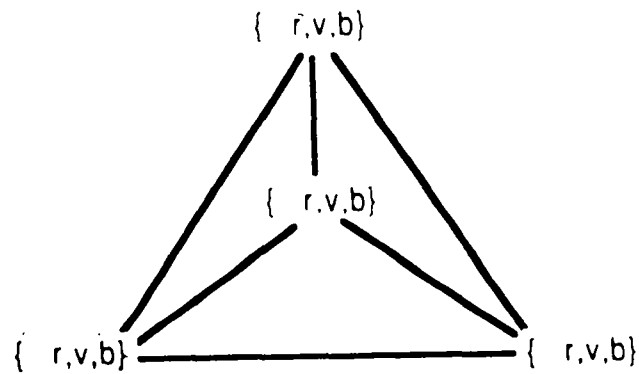


Figure 1: problème des 3-sommets

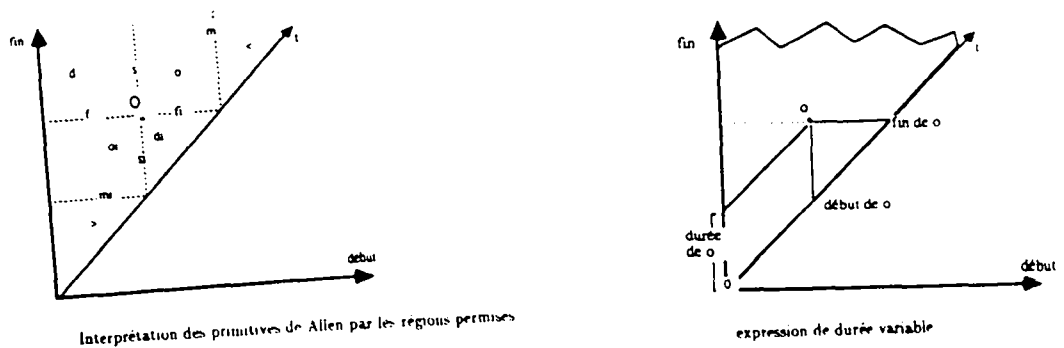


Figure 2: la représentation de Rit

l'ensemble de leurs occurrences possibles (SOPO). SOPO est le domaine continu d'un intervalle fermé dont le début et la fin sont connus. Les événements sont liés deux à deux par les relations de Allen. Le but de la propagation de contraintes (pour l'ordonnancement) est de modifier ces ensembles d'occurrences de telle sorte qu'ils deviennent compatibles avec les relations qui lient chaque couple d'événements qui agissent comme des contraintes sur les ensembles. Rit utilise une représentation bidimensionnelle du temps (cf figure 2). Son algorithme (algorithme de filtrage du type AC2 de Waltz) est de complexité n^2 . Puisque Rit utilise un domaine dense et continu, il a été obligé de considérer ses ensembles comme des ensembles fermés.

Résumons les problèmes dont souffre la propagation de contraintes :

- complexité élevée;
- problème de la cohérence globale;
- impossibilité de propager les contraintes symboliques et numériques en même temps.

Le paragraphe suivant introduit notre solution pour les problèmes mentionnés ci-dessus.

2 Une solution possible

Nos solutions sont fondées sur la spécification d'un module gestionnaire de graphe temporel. Cela implique la définition théorique d'une représentation des informations temporelles, et la propagation des contraintes.

Pour la représentation nous avons adopté un point de vue qualitatif du temps. Notre but final est d'aboutir à une représentation où les deux points de vue mentionnés auparavant, c'est-à-dire les instants et les intervalles, ainsi que les contraintes numériques et symboliques coopèrent. Nous avons choisi comme base de notre représentation les intervalles d'Allen.

Nous avons choisi d'étudier une représentation discrète du temps qui nous permet d'utiliser des algorithmes efficaces ainsi qu'une représentation très souple. Nous utilisons une représentation 2-D du temps. Sur l'axe horizontal on définit un référentiel temporel Tr . Celui-ci est divisé en unités de mesure u_i , les u_i étant les unités temporelles les plus petites possibles que l'on puisse manipuler. Les relations entre les unités temporelles peuvent être formulées en utilisant les relations *before* et *meets* de Allen:

$$\forall u_i \in Tr, \exists u_{i-1}, u_{i+1} \in Tr \text{ tel que } u_{i-1} < u_i < u_{i+1} \\ \text{et } ((u_{i-1} \text{ before } u_i) \vee (u_{i-1} \text{ meets } u_i)) \wedge ((u_i \text{ before } u_{i+1}) \vee (u_i \text{ meets } u_{i+1}))$$

Bien que Tr définisse un domaine discret du temps, nous pouvons représenter des intervalles ouverts.

Sur l'axe vertical sont représentés les événements. A chaque événement est associée une durée D spécifique, exprimée en nombre d'unités temporelles u , consécutives (la cardinalité de l'ensemble). Un événement a lieu dans une fenêtre W qui est un sous-ensemble du Tr . Chaque fenêtre est bornée par une date inférieure et une date supérieure (da_{inf} , da_{sup}). Les événements

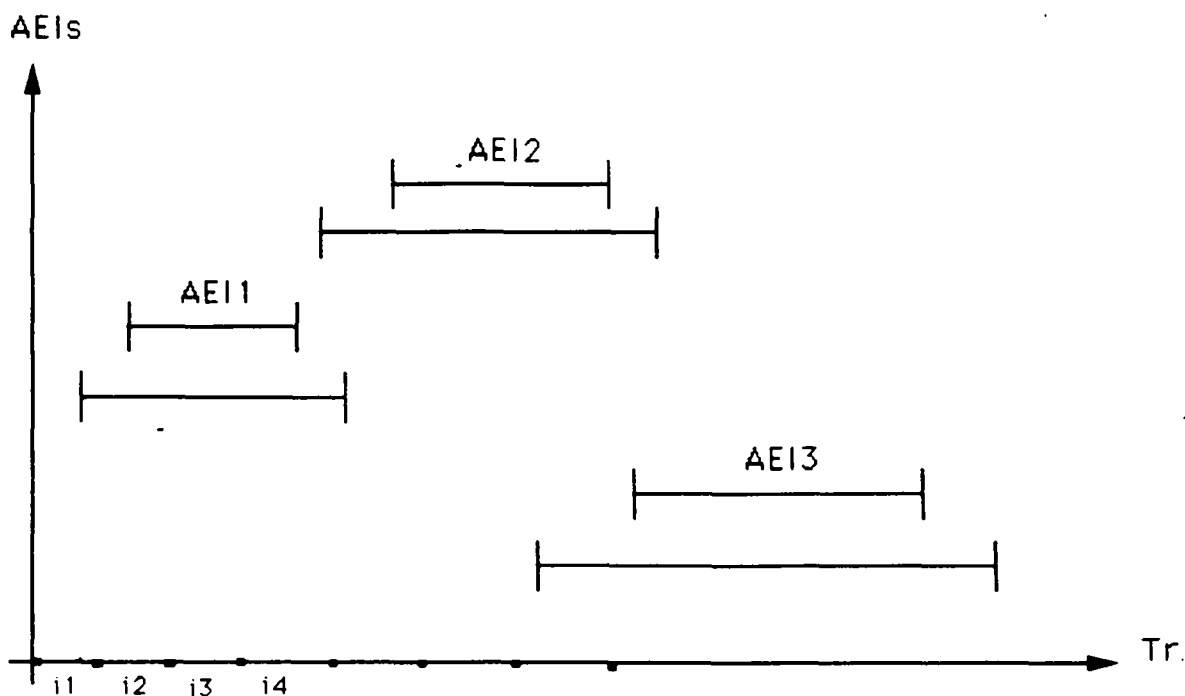


Figure 3: Représentation des événements

sont liés par des contraintes exprimées par un nombre quelconque de relations de Allen. Ainsi, chaque événement est contraint par sa fenêtre et sa durée. (cf. figure 3). Ce formalisme se représente facilement sous la forme d'un graphe temporel. La propagation de contraintes détermine les relations réelles entre les événements (positions relative représentée sous la forme de contraintes symboliques), et leurs dates (les contraintes numériques, pour l'ordonnancement) sur le référentiel temporel.

Du fait de la nature hétérogène des contraintes (symboliques et numériques) nous proposons une nouvelle stratégie pour la propagation de contraintes. Cette stratégie consiste en deux niveaux de propagation de contraintes. Le premier revient à propager les contraintes symboliques (propagateur symbolique) pour décider de la relativité entre les événements. A ce niveau, nous utilisons un algorithme défini sur l'ensemble des relations disjonctives avec une loi de composition interne (transitivité de Allen). Nous utilisons l'algorithme de consistance d'arcs GAC4 développé dans notre équipe [MM88]. Cet algorithme est la généralisation de l'algorithme AC4 [MM86]. Son intérêt réside dans le fait qu'il peut manipuler des contraintes *n-aires*. Dans ce cas, une contrainte représente une relation $R_{i,j,\dots,k}$ qui spécifie les étiquettes ad-

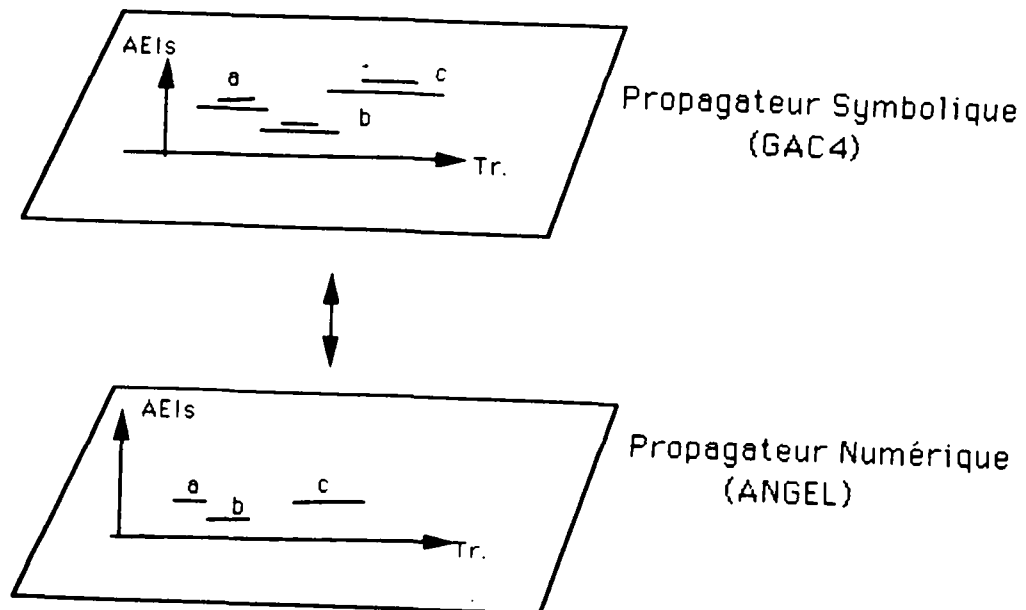


Figure 4: Propagation de contraintes à deux niveaux

missibles pour les nœuds $i, j, \dots k$. Elle est représentée par une arête dans un *hypergraphe*. On peut aussi la définir par l'énumération des ensembles d'étiquettes admissibles pour tous les n -uplets de nœuds $\{i, j, \dots k\}$: $((i, a), (j, b), \dots (k, c))$.

Pour le deuxième niveau de propagation des contraintes numériques, nous utilisons un algorithme que nous avons spécialement développé, ANGEL. ANGEL a par ailleurs la capacité de propager des contraintes n -aires comme GAC4. Après avoir établi toutes les positions relatives sur le référentiel temporel, ce niveau consiste à propager les contraintes numériques associées aux événements. L'algorithme précise les dates (sur Tr) pour chaque événement en respectant les contraintes symboliques par lesquelles il est lié aux autres événements et en éliminant les contraintes symboliques qui ne peuvent être satisfaites. (cf. figure 4). A la fin de ces deux étapes nous obtenons des événements ordonnés symboliquement entre eux avec leurs dates possibles sur le Tr .

3 Propagation de contraintes temporelles

Dans ce paragraphe nous présentons notre propagateur de contraintes à deux niveaux, avec les algorithmes et les résultats expérimentaux illustrant les améliorations apportées par rapport aux algorithmes dont nous avons montré les limites.

3.1 Propagateur symbolique

A ce niveau, le propagateur symbolique GAC4 travaille sur l'élagage récursif des étiquettes. Quand une étiquette a doit être éliminée de l'ensemble L_i des étiquettes admissibles pour le nœud i , tous les $n - \text{uplets}$ qui contiennent a doivent être supprimés. Dès qu'on supprime le dernier $n - \text{uplet}$ d'une relation R supportant une étiquette particulière cette dernière doit être supprimée, etc. Pour avoir un étiquetage de cohérence d'arc la condition suivante doit être respectée:

$$\forall i \in N, \forall a_i \in L_i, \forall R_{j,\dots,k} \text{ contraignant } i, \forall j, \dots, k, \exists a_j, \dots, a_k \text{ tel que } R_{j,\dots,k}(a_j, \dots, a_k) \text{ est vrai}$$

Un étiquetage est dit cohérent ssi:

$$\forall i \in N, \forall a_i \in L_i, \forall j \in N, \exists a_j \in L_j \text{ tel que } \forall R_{k,\dots,j}, R_{k,\dots,j}(a_k, \dots, a_j) \text{ est vrai}$$

Pour chaque étiquetage (i, a) , il existe une autre étiquette le supportant sur un autre nœud, de telle sorte que les contraintes soient simultanément satisfaites (localement cohérentes). Montanari [Mon74] a introduit une condition pour la consistance de chemins qui est plus forte que celle de la cohérence locale:

$$\begin{aligned} & \forall i, j \in N, \forall a \in L_i, \forall b \in L_j, R_{i,j}(a, b) \text{ est vrai ssi} \\ & \forall k \in N, \exists c \in L_k \text{ tel que } R_{i,k}(a, c) \text{ et } R_{k,j}(c, b) \text{ sont vrais} \end{aligned}$$

Cette condition porte sur un triplet de nœuds. Ainsi, après une simple transformation de notre réseau de contraintes, GAC4 peut manipuler la condition de consistance de chemins. Cette transformation consiste à construire un autre réseau $A(N)$ à partir du réseau original N , où dans $A(N)$ chaque étiquette est le couple (i, j) de N . Les étiquetages possibles sont les paires

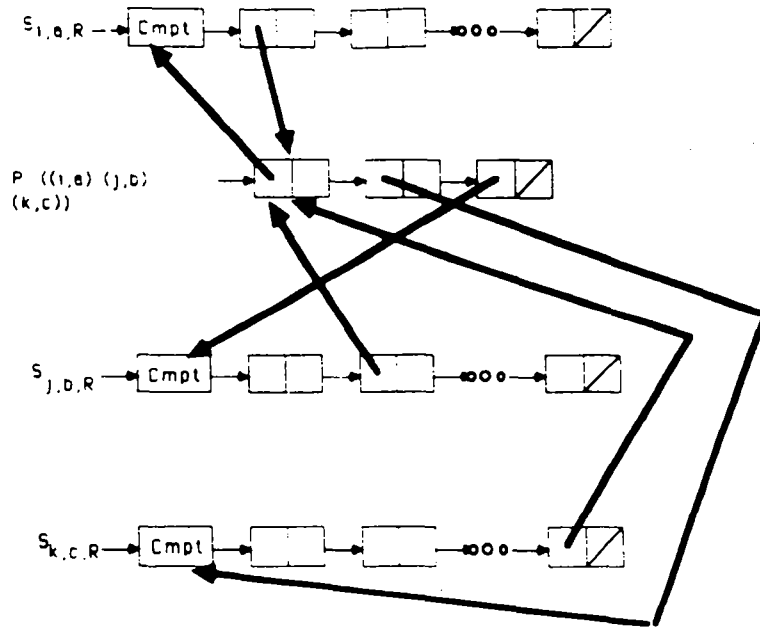


Figure 5: Structure de données dans GAC4

(a, b) , où $a \in L_i$ et $b \in L_j$. En utilisant les contraintes R du réseau initial N on introduit les contraintes R^3 sur chaque triplet de nœuds:

$$R^3_{(i,j),(j,k),(k,i)}((a,b), (b,c), (c,a)) \text{ est vrai}$$

ssi $R_{i,j}(a,b)$ et $R_{j,k}(b,c)$ et $R_{k,i}(c,a)$ sont vrais

N possède donc la propriété de cohérence de chemins ssi $A(N)$ a la cohérence d'arcs pour cette relation ternaire. Une preuve complète se trouve dans [Moh87]. GAC4 est fondé sur la notion de support, c'est pourquoi avant de commencer l'étape d'élagage il faut construire la bonne structure de données. Pour cette raison et pour avoir un accès rapide aux étiquettes à éliminer, la première étape est celle où nous construisons la structure de données dite liste chaînée bilatérale (cf figure 5). La notion de support est représentée par des compteurs donnant les nombres de supports existants pour chaque étiquette admissible en R . Dès qu'un compteur arrive à zéro il faut éliminer l'étiquette car il n'y a pas d'autre étiquette sur les autres nœuds supportant son existence. La deuxième étape consiste en un élagage au cours duquel les étiquettes que ne sont pas admissibles sont supprimées. Les deux étapes sont totalement indépendantes.

GAC4 Temporel

Afin de respecter la condition de cohérence globale nous avons introduit dans l'initialisation une étape de prétraitement au cours de laquelle on transforme le graphe temporel G dont les nœuds sont des événements et les arcs les relations de Allen en un graphe $A'(G)$ où chaque arc représente un événement et où les nœuds sont les relations de Allen qui contraignent cet événement. Après avoir terminé cette étape, on calcule pour chaque triplet de nœuds (i, j, k) de $A'(G)$ la condition $R_{i,j,k}^3$: l'ensemble des triplets d'étiquettes valides pour l'*hyper-arc* (i, j, k) . En appliquant une simple opération de différence entre l'ensemble obtenu précédemment et l'ensemble des étiquettes impossibles résultant d'une opération similaire sur toutes les 13^3 combinaisons de Allen et son tableau de transitivité (409 entrées), on obtient toutes les étiquettes impossibles $IL_{i,j,k}$ pour l'*hyper-arc* (i, j, k) . L'étape d'initialisation construit la structure de données en utilisant $IL_{i,j,k}$ comme étiquettes qui doivent être éliminées en premier. De ce fait, le problème des trois sommets n'existe pas car avec le même principe et en utilisant la loi de transitivité de van Beek [vB89], GAC4 peut manipuler des quadruplets de nœuds au lieu de triplets.

Complexité

La complexité et le temps d'exécution des algorithmes de Allen, de van Beek et GAC4 ont été calculés. Le tableau suivant donne une comparaison des complexités, où n est le nombre de nœuds du graphe G .

| Algorithme | Complexité |
|-----------------------|------------|
| Allen | n^3 |
| van Beek (one to all) | n^3 |
| van Beek (all to all) | n^4 |
| GAC4 | $n^3/2$ |

Pour le temps d'exécution, les algorithmes ont été codés en *Common Lisp* et tournent sous UNIX sur poste de travail SUN-3. Notons que GAC4 travaille sur les *hyper-arcs*.

| Algorithme | nombre de nœuds | temps (en secondes) |
|--------------------------|-----------------|---------------------|
| Allen | 10 | 44.66 |
| | 20 | 187.37 |
| van Beek (one to all) | 10 | 86.78 |
| | 20 | 1101.80 |
| van Beek (all to all) | 10 | 44.79 |
| | 20 | 351.93 |

Pour GAC4:

| nombre de nœuds | nombre d'hyper-arcs | étape d'initialisation (en secondes) | étape d'élagage (en secondes) |
|-----------------|---------------------|---|----------------------------------|
| 31 | 18 | 18.00 | 0.60 |
| 40 | 30 | 25.08 | 1.44 |
| 75 | 55 | 37.97 | 3.69 |
| | 720 | les deux étapes: 295.38 | |

3.2 Propagateur numérique

L'étape précédente a établi les positions relatives entre les événements. Dans cette étape on utilise les contraintes numériques sous la forme de fenêtres et de durées pour préciser les dates exactes de chaque événement, et éliminer les contraintes symboliques qui ne sont pas satisfaites globalement.

Nous utilisons l'algorithme de consistance d'arcs ANGEL [TM90]. ANGEL tourne sur le graphe construit lors de la première étape. Les mêmes principes et conditions pour la cohérence globale mentionnés ci-dessus sont appliqués ici.

ANGEL est un algorithme dynamique, c'est-à-dire qu'il a la capacité de construire les graphes sur lesquels il travaille en considérant les contraintes une par une. Tandis que les entrées de GAC4 sont les ensembles de relations non admissibles obtenues à l'étape d'initialisation, ANGEL reçoit l'ensemble des relations admissibles (l'étape d'initialisation étant évitée). Les contraintes sont placées dans une file par ordre d'arrivée. Si l'hyper-arc correspondant à la contrainte courante n'existe pas il est créé. Les dates de cet arc sont ajustées, si nécessaire, par rapport aux contraintes symboliques et les contraintes symboliques non satisfaites sont éliminées. Si les dates d'un hyper-arc sont changées, tous les hyper-arcs ayant un arc commun avec celui-ci sont ajoutés à la file afin de les examiner plus tard. Dès que la file est vide,

ANGEL s'arrête en attente d'une nouvelle entrée. Notre algorithme possède une architecture assez simple : il dispose d'une première file qui ressemble à une fenêtre ouverte sur le monde extérieur et d'une seconde appartenant au corps de l'algorithme. Par défaut, la seconde est de priorité supérieure. Il dispose également d'un archivage où sont sauvegardés un historique des hyper-arcs traités ainsi que l'architecture du graphe (cf figure 6).

3.3 Complexité

Puisque nous avons un nombre fini d'étiquettes et d'hyper-arcs, la terminaison de l'algorithme ANGEL est garantie. Pour la complexité, on considère un graphe NN où le nombre de noeuds n est égal au nombre d'hyper-arcs de notre graphe N . La complexité pour NN est alors équivalente à la complexité d'un problème de fermeture transitive de l'ordre de n^3 . Donc, pour un triplet de noeuds (i, j, k) (en N) la complexité est égale à $\frac{n^3}{3}$. Le tableau ci-dessous donne une comparaison avec l'algorithme de filtrage de Rit:

| Rit (AC2) | ANGEL |
|------------------------------|-------------------------------|
| n^2 (n nombre de noeuds) | n^3 (n nombre d'hyper-arcs) |
| Statique | Dynamique |
| Contraintes binaires | contraintes n-aires |
| utilise un domaine continu | utilise un domaine discret |
| Assure la consistance locale | Assure la consistance globale |
| Problème fixe | Incrémental |

4 Perspectives et conclusion

Nous avons présenté dans cet article une approche pour la représentation des informations temporelles permettant la coopération entre informations numériques et symboliques. Nous proposons une stratégie originale de propagation de contraintes à deux niveaux, symbolique et numérique, avec une cohérence globale assurée à chaque niveau. Nous avons également décrit les algorithmes correspondants, ANGEL et GAC4. Notre algorithme de propagation de contraintes ANGEL est dynamique et peut être utilisé en interaction avec un moteur d'inférences. La complexité de nos algorithmes se compare favorablement aux algorithmes équivalents connus dans le domaine.

```

Tant que Queue (Q)  $\neq$  0 faire
    dépiler Q et prendre un hyper-arc Ha
    Si Ha n'a pas été déjà traité
    alors
        mettre à jour le graphe (architecture et domaines)
        fin si.
        Appliquer les contraintes sur les domaines d'hyper-arc
        Si les domaines sont changés
        alors
            empiler tous les hyper-arcs ayant un arc en commun avec
            celui en question pour être examinés plus tard
            fin si.
        fin tant que.

```

Figure 6: L'algorithme ANGEL

Les travaux actuels portent en parallèle sur les deux aspects du problème, la représentation des connaissances temporelles (en particulier, la possibilité de manipuler plusieurs granularités du temps) et la propagation de contraintes (amélioration des algorithmes existants, interface avec un moteur d'inférences et contrôle intelligent du gestionnaire de graphe temporel). Ces travaux fondamentaux sont actuellement en cours d'application dans le cadre d'un projet du raisonnement temporel pour la conduite de procédés industriels.

References

- [All83] J. F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of ACM*, 26(11):832-843, nov. 1983.

- [BT84] C. E. Bell and A. Tate. Using temporal constraints to restrict search in a planner. Technical report, AI application institute - Univ. of Edinbourg, December 1984. presented to the 3ed W.S. of UK Alvey IKBS programmés.
- [DeK86] J. DeKleer. An Assumption-based TMS. *AI*, 28:127-162, 1986.
- [DM87] T. L. Dean and D. V. McDermott. Temporal Data Base Management. *AI*, 32:1-55, 1987.
- [Doy79] J. Doyle. A Truth Maintenance System. *AI*, 12:231-272, 1979.
- [KS86] R. Kowalski and M. Segot. A Logic Based Calculus of Events. *New Generation Computing*, 4:67 - 95, 1986.
- [MB83] J. Malik and T. O. Binford. Reasoning in Time and Space. In *IJCAI*, pages 342-345, 1983.
- [McD82] D.V. McDermott. A Temporal Logic for Reasoning about Processes and Plans. *Cognitive Science*, 6:101-155, 1982.
- [MM86] R. Mohr and G. Masini. Running Efficiently Arc Consistency. In *ECAI*, Lifa Grnoble and CRIN, 1986.
- [MM88] R. Mohr and G. Masini. Good Old Discrete Relaxation. In *9th European conf on AI (ECAI)*, pages 651-656, 1988. munich.
- [Moh87] R. Mohr. A Correct Path Consistency and an Optimal Generalized Arc Consistency Algorithm. Rapport interne, CRIN-INRIA Lorraine, 1987.
- [Mon74] U. Montanari. Networks of Constraints: Fundamental Properties and Applications to Picture Processing. *Information Sciences*, 7:95-132, 1974.
- [Rit86] J.F. Rit. Propagating Temporal Constraints For Scheduling. In *AAAI*, volume 1, pages 383-388, 1986.
- [Rit88] J.F. Rit. *Modélisation et Propagation de Contraintes Temporelles Pour La Planification*. PhD thesis, INPL, 7 mars 1988.

- [Sho87] Y. Shoham. Temporal logics in AI: Semantical and Ontological considerations. *AI*, 33:89–104, 1987.
- [Tat77] A. Tate. Generating Project Networks. In *IJCAI*, volume 2, pages 888–893, 1977.
- [TM90] H. Tolba and G. Masini. Temporal Constraint Propagation. Technical report, CRIN 90-R-177, CRIN-INRIA, 1990.
- [vB89] P. van Beek. Approximation Algorithms for Temporal Reasoning. In *IJCAI*, pages 1291–1296, 1989.
- [Ver83] S. A. Vere. Planning in time: Windows and durations for activities and goals. *IEEE trans. on PAMI*, PAMI:5(3):246–267, May 1983.
- [VK86] M. Vilain and H. Kautz. Constraint Propagation Algorithms for Temporal Reasoning. In *AAAI*, pages 377–382, 1986.

ISSN 0249 - 6399